

LISTING OF CLAIMS:

The following listing of claims will replace all prior versions, and listings, of claims in the application.

1. (Previously Presented) A computer-implemented method for creating a graphical program, the method comprising:

creating a graphical user interface for the graphical program in response to first user input;

displaying a first node for receiving user interface events in a block diagram for the graphical program in response to second user input;

receiving third user input explicitly specifying one or more user interface events to configure for the first node;

configuring the first node to receive the one or more user interface events explicitly specified by the third user input during execution of the graphical program; and

associating one or more portions of graphical code with the first node in response to fourth user input, wherein each portion of graphical code comprises one or more nodes for responding to one or more of the user interface events which the first node is configured to receive.

2. (Previously Presented) The method of claim 1,

wherein the first node comprises one or more sub-diagrams, wherein said associating the one or more portions of graphical code with the first node comprises displaying each portion of graphical code within one of the sub-diagrams of the first node.

3. (Previously Presented) The method of claim 2,

wherein said receiving the third user input explicitly specifying the one or more user interface events to configure for the first node comprises receiving user input explicitly specifying one or more user interface events to which each of the sub-diagrams of the first node corresponds;

wherein for each portion of graphical code, displaying the portion of graphical code within one of the sub-diagrams of the first node comprises configuring the portion of graphical code to execute in response to the one or more user interface events to which the sub-diagram corresponds.

4. (Previously Presented) The method of claim 3,

wherein for each portion of graphical code, displaying the portion of graphical code within one of the sub-diagrams of the first node comprises displaying the one or more nodes of the portion of graphical code within one of the sub-diagrams of the first node.

5. (Original) The method of claim 1,

wherein the block diagram comprises a data flow block diagram.

6. (Previously Presented) The method of claim 1, further comprising:

executing the graphical program;

wherein one or more user interface events which the first node is configured to receive are generated during execution of the graphical program;

wherein the method further comprises executing one of the portions of graphical code associated with the first node in response to each of the one or more user interface events which the first node is configured to receive being generated during execution of the graphical program.

7. (Original) The method of claim 6,

wherein the one or more user interface events generated during execution of the graphical program are generated in response to user input to the graphical user interface of the graphical program.

8. (Original) The method of claim 6,

wherein, during execution of the graphical program, the block diagram executes on a first computer system and the graphical user interface is displayed on a display of a second computer system.

9. (Original) The method of claim 6,

wherein, during execution of the graphical program, the graphical user interface is displayed on a display of a computer system and the block diagram executes on a reconfigurable instrument connected to the computer system.

10. (Previously Presented) The method of claim 1,

wherein said configuring the first node to receive the one or more user interface events comprises configuring the first node to receive notification when the one or more user interface events are generated during execution of the graphical program.

11. (Previously Presented) The method of claim 1,

wherein said configuring the first node to receive the one or more user interface events comprises configuring the first node to receive information specifying occurrences of the one or more user interface events during execution of the graphical program.

12. (Previously Presented) The method of claim 1,

wherein each of the portions of graphical code associated with the first node is displayed within the first node.

13. (Previously Presented) The method of claim 1,

wherein said configuring the first node to receive the one or more user interface events comprises configuring the first node to receive a first user interface event;

wherein the first user interface event explicitly specifies a first user interface element of the graphical user interface and an action performed on the first user interface element.

14. (Original) The method of claim 13, wherein the first user interface element comprises one of:

- an indicator;
- a control;
- a menu element;
- a window.

15. (Previously Presented) The method of claim 1, further comprising:
displaying a first graphical user interface dialog for configuring the first node;
wherein said receiving the third user input explicitly specifying the one or more user interface events to configure for the first node comprises receiving user input to the first graphical user interface dialog to explicitly specify the one or more user interface events.

16. (Currently Amended) The method of claim 1, further comprising:
displaying a second node for dynamically registering user interface events in the block diagram in response to user input;

wherein said receiving the third user input explicitly specifying the one or more user interface events to configure for the first node comprises receiving user input explicitly specifying a first user interface event to dynamically register during execution of the graphical program;

wherein the method further comprises configuring the second node to dynamically register the first user interface event during execution of the graphical program such that, after registering the event, the first node becomes operable to receive the first user interface event. [[:]]

~~wherein, before said dynamically registering the first user interface event, the first node is not operable to receive the first user interface event;~~

~~wherein, after said dynamically registering the first user interface event, the first node is operable to receive the first user interface event.~~

17. (Previously Presented) The method of claim 16,

wherein said configuring the second node to dynamically register the first user interface event during execution of the graphical program comprises connecting the second node to the first node in response to user input.

18. (Previously Presented) The method of claim 1,

wherein the one or more user interface events specified by the third user input includes a first user interface event;

wherein the method further comprises displaying a second node for dynamically un-registering user interface events in the block diagram in response to user input;

configuring the second node to dynamically un-register the first user interface event during execution of the graphical program;

wherein, before said dynamically un-registering the first user interface event, the first node is operable to receive the first user interface event;

wherein, after said dynamically un-registering the first user interface event, the first node is not operable to receive the first user interface event.

19. (Previously Presented) A computer-implemented method for creating a graphical program, the method comprising:

displaying a first node for receiving programmatic events in a block diagram for the graphical program in response to first user input;

receiving second user input explicitly specifying one or more programmatic events to configure for the first node;

configuring the first node to receive the one or more programmatic events explicitly specified by the second user input during execution of the graphical program; and

associating one or more portions of graphical code with the first node in response to third user input, wherein each portion of graphical code comprises one or more nodes for responding to one or more of the programmatic events which the first node is configured to receive.

20. (Previously Presented) The method of claim 19,

wherein the first node comprises one or more sub-diagrams, wherein said associating the one or more portions of graphical code with the first node comprises displaying each portion of graphical code within one of the sub-diagrams of the first node.

21. (Previously Presented) The method of claim 20,

wherein said receiving the second user input explicitly specifying the one or more programmatic events to configure for the first node comprises receiving user input explicitly specifying one or more programmatic events to which each of the sub-diagrams of the first node corresponds;

wherein for each portion of graphical code, displaying the portion of graphical code within one of the sub-diagrams of the first node comprises configuring the portion of graphical code to execute in response to the one or more user interface events to which the sub-diagram corresponds.

22. (Original) The method of claim 19, wherein the one or more programmatic events comprise one or more of:

a user interface event;

a system event;

a timer event;

an event generated in response to data acquired from a device.

23. (Previously Presented) A computer-implemented method for creating a graphical program, the method comprising:

creating a graphical user interface for the graphical program in response to user input;

creating a block diagram for the graphical program in response to user input, wherein creating the block diagram comprises creating a first portion of graphical code in response to user input, wherein the first portion of graphical code includes one or more nodes for responding to user interface events;

receiving user input explicitly specifying one or more user interface events to associate with the first portion of graphical code; and

configuring the first portion of graphical code to execute in response to the one or more explicitly specified user interface events being generated during execution of the graphical program.

24. (Previously Presented) The method of claim 23,

wherein the first portion of graphical code includes a plurality of nodes, wherein said creating the first portion of graphical code in response to user input comprises arranging the plurality of nodes on a display and interconnecting the plurality of nodes in response to user input, wherein the plurality of interconnected nodes visually indicates functionality for responding to the one or more user interface events associated with the first portion of graphical code.

25. (Previously Presented) The method of claim 23, further comprising:

executing the graphical program;

wherein one or more user interface events associated with the first portion of graphical code are generated during execution of the graphical program;

wherein the method further comprises executing the first portion of graphical code in response to the one or more user interface events associated with the first portion of graphical code being generated.

26. - 31. (Canceled)

32. (Previously Presented) A memory medium for creating a graphical program, the memory medium comprising program instructions executable to:

create a graphical user interface for the graphical program in response to first user input;

display a first node for receiving user interface events in a block diagram for the graphical program in response to second user input;

receive third user input explicitly specifying one or more user interface events to configure for the first node;

configure the first node to receive the one or more user interface events explicitly specified by the third user input during execution of the graphical program; and

associate one or more portions of graphical code with the first node in response to fourth user input, wherein each portion of graphical code comprises one or more nodes for responding to one or more of the user interface events which the first node is configured to receive.

33. (Previously Presented) The memory medium of claim 32,

wherein the first node comprises one or more sub-diagrams, wherein said associating the one or more portions of graphical code with the first node comprises displaying each portion of graphical code within one of the sub-diagrams of the first node.

34. (Previously Presented) The memory medium of claim 33,

wherein said receiving the third user input explicitly specifying the one or more user interface events to configure for the first node comprises receiving user input explicitly specifying one or more user interface events to which each of the sub-diagrams of the first node corresponds;

wherein for each portion of graphical code, displaying the portion of graphical code within one of the sub-diagrams of the first node comprises configuring the portion of graphical code to execute in response to the one or more user interface events to which the sub-diagram corresponds.

35. (Previously Presented) The memory medium of claim 34,

wherein for each portion of graphical code, displaying the portion of graphical code within one of the sub-diagrams of the first node comprises displaying the one or more nodes of the portion of graphical code within one of the sub-diagrams of the first node.

36. (Previously Presented) A computer-implemented method for creating a graphical program, the method comprising:

displaying a first node in a block diagram of the graphical program in response to user input;

associating a first portion of graphical source code with the first node in response to user input, wherein the first portion of graphical source code associated with the first node comprises a plurality of interconnected nodes;

associating a first user interface event with the first node in response to user input explicitly specifying the first user interface event; and

configuring the first portion of graphical source code associated with the first node to execute in response to the first user interface event associated with the first node being generated during execution of the graphical program.

37. (Previously Presented) The method of claim 36,

wherein said first portion of graphical source code executing in response to the first user interface event comprises the first portion of graphical source code executing in response to the first user interface event during execution of the graphical program, wherein the first user interface event is generated during execution of the graphical program.

38. (Previously Presented) The method of claim 36,

wherein the plurality of interconnected nodes of the first portion of graphical source code associated with the first node visually indicates functionality for responding to the first user interface event associated with the first node.

39. (Previously Presented) The method of claim 36, further comprising:

creating the first portion of graphical source code in response to user input, wherein creating the first portion of graphical source code comprises:

displaying the plurality of nodes in response to user input; and

interconnecting the plurality of nodes in response to user input.

40. (Previously Presented) The method of claim 36,
wherein said associating the first portion of graphical source code with the first node in response to user input comprises displaying the first portion of graphical source code within the first node in response to user input, wherein the first portion of graphical source code associated with the first node is visible in the block diagram of the graphical program.

41. (Previously Presented) The method of claim 36, further comprising:
displaying information explicitly identifying a plurality of user interface events;
wherein said associating the first user interface event with the first node in response to user input explicitly specifying the first user interface event comprises receiving user input selecting the first user interface event from the displayed information.

42. (Previously Presented) The method of claim 36, further comprising:
associating a second portion of graphical source code with the first node in response to user input;
associating a second user interface event with the first node in response to user input explicitly specifying the second user interface event; and
configuring the second portion of graphical source code associated with the first node to execute in response to the second user interface event associated with the first node.

43. (Previously Presented) The method of claim 36,
wherein the user input explicitly specifying the first user interface event comprises user input explicitly specifying a name of the first user interface event.

44. (Previously Presented) The method of claim 36, further comprising:
displaying a graphical user interface dialog, wherein the graphical user interface dialog explicitly displays a plurality of user interface events;

wherein said receiving user input explicitly specifying the first user interface event comprises receiving user input via the graphical user interface dialog to select the first user interface event from the plurality of explicitly displayed user interface events.

45. (Canceled).

46. (Previously Presented) The method of claim 36,
wherein the graphical program has a graphical user interface;
wherein the first user interface event is associated with a first user interface element of the graphical user interface.

47. (Previously Presented) The method of claim 46, wherein the first user interface element comprises one of:

- an indicator;
- a control;
- a menu element;
- a window.

48. (Previously Presented) The method of claim 36,
wherein the graphical program includes a graphical user interface;
wherein the first user interface event is associated with a user action performed on the graphical user interface.

49. (Previously Presented) The method of claim 36, further comprising:
executing the graphical program;
generating the first user interface event during execution of the graphical program; and
executing the first portion of graphical source code associated with the first node in response to said generating the first user interface event.

50. (Previously Presented) The method of claim 36,

wherein the graphical program includes a graphical user interface;

wherein the method further comprises:

executing the graphical program;

generating the first user interface event during execution of the graphical program, wherein said generating the first user interface event comprises generating the first user interface event in response to user input to the graphical user interface; and

executing the first portion of graphical source code associated with the first node in response to said generating the first user interface event.

51. (Previously Presented) The method of claim 36,

wherein the block diagram of the graphical program comprises a data flow diagram.

52. (Previously Presented) The method of claim 36,

wherein the block diagram comprises a plurality of interconnected nodes that visually indicate functionality of the graphical program.

53. (Previously Presented) A memory medium for creating a graphical program, the memory medium comprising program instructions executable to:

display a first node in a block diagram of the graphical program in response to user input;

associate a first portion of graphical source code with the first node in response to user input, wherein the first portion of graphical source code associated with the first node comprises a plurality of interconnected nodes;

associate a first user interface event with the first node in response to user input explicitly specifying the first user interface event; and

configure the first portion of graphical source code associated with the first node to execute in response to the first user interface event associated with the first node being generated during execution of the graphical program.

54. (Previously Presented) The memory medium of claim 53,

wherein said first portion of graphical source code executing in response to the first user interface event comprises the first portion of graphical source code executing in response to the first user interface event during execution of the graphical program, wherein the first user interface event is generated during execution of the graphical program.

55. (Previously Presented) The memory medium of claim 53, wherein the plurality of interconnected nodes of the first portion of graphical source code associated with the first node visually indicates functionality for responding to the first user interface event associated with the first node.

56. (Previously Presented) The memory medium of claim 53, wherein the program instructions are further executable to:

create the first portion of graphical source code in response to user input, wherein creating the first portion of graphical source code comprises:

displaying the plurality of nodes in response to user input; and
interconnecting the plurality of nodes in response to user input.

57. (Previously Presented) The memory medium of claim 53, wherein said associating the first portion of graphical source code with the first node in response to user input comprises displaying the first portion of graphical source code within the first node in response to user input, wherein the first portion of graphical source code associated with the first node is visible in the block diagram of the graphical program.

58. (Previously Presented) The memory medium of claim 53, wherein the program instructions are further executable to:

display information explicitly identifying a plurality of user interface events;

wherein said associating the first user interface event with the first node in response to user input explicitly specifying the first user interface event comprises

receiving user input selecting the first user interface event from the displayed information.

59. (Previously Presented) The memory medium of claim 53, wherein the program instructions are further executable to:

associate a second portion of graphical source code with the first node in response to user input;

associate a second user interface event with the first node in response to user input explicitly specifying the second user interface event; and

configure the second portion of graphical source code associated with the first node to execute in response to the second user interface event associated with the first node.

60. (Previously Presented) The memory medium of claim 53,

wherein the memory medium further comprises program instructions executable to display a graphical user interface dialog, wherein the graphical user interface dialog explicitly displays a plurality of user interface events;

wherein said receiving user input explicitly specifying the first user interface event comprises receiving user input via the graphical user interface dialog to select the first user interface event from the plurality of explicitly displayed user interface events.

61. (Canceled)

62. (Previously Presented) The memory medium of claim 53,

wherein the graphical program includes a graphical user interface;

wherein the first user interface event is associated with a first user interface element of the graphical user interface.

63. (Previously Presented) The memory medium of claim 53,

wherein the graphical program includes a graphical user interface;

wherein the first user interface event is associated with a user action performed on the graphical user interface.

64. (Previously Presented) The memory medium of claim 53,
wherein the block diagram of the graphical program comprises a data flow diagram.

65. (Previously Presented) The memory medium of claim 53,
wherein the block diagram comprises a plurality of interconnected nodes that visually indicate functionality of the graphical program.

66. (Previously Presented) A system for creating a graphical program, the system comprising:

- a memory storing program instructions;
- a processor coupled to the memory; and
- a display device;

wherein the processor is operable to execute the program instructions stored in the memory to:

display a first node on the display device in response to user input, wherein said displaying the first node comprises displaying the first node in a block diagram of the graphical program;

associate a first portion of graphical source code with the first node in response to user input, wherein the first portion of graphical source code associated with the first node comprises a plurality of interconnected nodes;

associate a first user interface event with the first node in response to user input explicitly specifying the first user interface event; and

configure the first portion of graphical source code associated with the first node to execute in response to the first user interface event associated with the first node being generated during execution of the graphical program.

67. (Previously Presented) The method of claim 1, further comprising:

displaying a list of user interface events;

wherein said receiving the third user input explicitly specifying the one or more user interface events to configure for the first node comprises receiving user input to select the one or more user interface events from the displayed list of user interface events.

68. (Previously Presented) The method of claim 1,

wherein said receiving the third user input explicitly specifying the one or more user interface events to configure for the first node comprises receiving user input specifying names of the one or more user interface events to configure for the first node.

69. (Previously Presented) The method of claim 1,

wherein said associating the one or more portions of graphical code with the first node comprises associating a first portion of graphical code and a second portion of graphical code with the first node;

wherein the first node comprises a plurality of sub-diagrams;

wherein associating the first portion of graphical code with the first node comprises displaying the first portion of graphical code within a first sub-diagram of the first node; and

wherein associating the second portion of graphical code with the first node comprises displaying the second portion of graphical code within a second sub-diagram of the first node.

70. (Previously Presented) The method of claim 1, further comprising:

displaying one or more of the portions of graphical code associated with the first node in the block diagram of the graphical program.

71. (Currently Amended) The method of claim 1,

wherein said receiving the third user input explicitly specifying the one or more user interface events to configure for the first node comprises receiving user input

explicitly specifying a first user interface event to be dynamically registered during execution of the graphical program;

wherein the method further comprises configuring the graphical program to dynamically register the first user interface event during execution of the graphical program;

~~wherein, before said dynamically registering the first user interface event during execution of the graphical program, the first node is not operable to receive the first user interface event;~~

wherein said dynamically registering the first user interface event enables the first node to receive the first user interface event.

72. (Previously Presented) The method of claim 1,

wherein said receiving the third user input explicitly specifying the one or more user interface events to configure for the first node comprises receiving user input explicitly specifying one or more user interface events to which each of the portions of graphical code associated with the first node corresponds.

73. (Previously Presented) The method of claim 1,

wherein each of the portions of graphical code is comprised within the first node.

74. (Previously Presented) The method of claim 1,

wherein each of the portions of graphical code is located separately from the first node.

75. (Previously Presented) The method of claim 1,

wherein for each of the portions of graphical code, said associating the portion of graphical code with the first node comprises associating the portion of graphical code with one or more of the user interface events which the first node is configured to receive, wherein the portion of graphical code comprises one or more nodes for responding to the one or more user interface events with which the portion of graphical code is associated.

76. (Previously Presented) The method of claim 1,

wherein said receiving third user input explicitly specifying the one or more user interface events to configure for the first node comprises receiving user input explicitly specifying a first user interface event and a second user interface event;

wherein said associating the one or more portions of graphical code with the first node comprises associating a first portion of graphical code with the first node, wherein the first portion of graphical code comprises a plurality of interconnected nodes visually indicating functionality for responding to the first user interface event;

wherein said associating the one or more portions of graphical code with the first node further comprises associating a second portion of graphical code with the first node, wherein the second portion of graphical code comprises a plurality of interconnected nodes visually indicating functionality for responding to the second user interface event.

77. (Previously Presented) The method of claim 76, further comprising:

executing the graphical program;

executing the first portion of graphical code associated with the first node in response to the first user interface event being generated during execution of the graphical program; and

executing the second portion of graphical code associated with the first node in response to the second user interface event being generated during execution of the graphical program.

78. (Previously Presented) The method of claim 1, further comprising:

displaying information explicitly identifying a plurality of user interface events;

wherein said receiving the third user input explicitly specifying the one or more user interface events to configure for the first node comprises receiving user input selecting the one or more user interface events from the displayed information.

79. (Previously Presented) The method of claim 1,

wherein said receiving the third user input explicitly specifying the one or more user interface events to configure for the first node comprises graphically connecting a plurality of objects to the first node in response to user input, wherein each object directly represents a user interface event.

80. (Previously Presented) The method of claim 1,

wherein said receiving the third user input explicitly specifying the one or more user interface events to configure for the first node does not include receiving user input specifying a connection between the first node and a second node.

81. (Previously Presented) The method of claim 1,

wherein said receiving the third user input explicitly specifying the one or more user interface events to configure for the first node is performed independently of configuring other nodes in the block diagram of the graphical program.

82. (Previously Presented) The method of claim 1, further comprising:

creating the portions of graphical code in response to user input, wherein for each portion of graphical code, creating the portion of graphical code comprises including one or more nodes in the portion of graphical code in response to user input, wherein the one or more nodes are operable to respond to one or more of the user interface events which the first node is configured to receive.

83. (Previously Presented) The method of claim 23, further comprising:

displaying information explicitly identifying a plurality of user interface events;

wherein said receiving the user input explicitly specifying the one or more user interface events to associate with the first portion of graphical code comprises receiving user input selecting the one or more user interface events from the displayed information.

84. (Previously Presented) The method of claim 23, further comprising:

displaying a list of user interface events;

wherein said receiving the user input explicitly specifying the one or more user interface events to associate with the first portion of graphical code comprises receiving user input selecting the one or more user interface events from the displayed list of user interface events.

85. (Previously Presented) The method of claim 23,

wherein said receiving the user input explicitly specifying the one or more user interface events to associate with the first portion of graphical code comprises receiving user input specifying names of the one or more user interface events to associate with the first portion of graphical code.

86. (Previously Presented) A computer-implemented method for creating a graphical program, the method comprising:

displaying a first node for receiving programmatic events in a block diagram of the graphical program in response to user input;

associating a plurality of portions of graphical code with the first node in response to user input, wherein each portion of graphical code comprises one or more nodes;

receiving user input explicitly specifying one or more programmatic events to which each of the portions of graphical code associated with the first node corresponds; and

for each of the portions of graphical code associated with the first node, configuring the portion of graphical code to execute in response to the one or more programmatic events to which the portion of graphical code corresponds being generated during execution of the graphical program.

87. (Currently Amended) A computer-implemented method for creating a graphical program, the method comprising:

creating a first portion of graphical code in response to user input, wherein the first portion of graphical code comprises one or more nodes that visually indicate functionality for responding to programmatic events generated during execution of the graphical program; and

configuring the graphical program to dynamically register a first programmatic event during execution of the graphical program, wherein dynamically registering the first programmatic event comprises dynamically associating the first programmatic event with the first portion of graphical code;

~~wherein, before said dynamically registering the first programmatic event during execution of the graphical program, the first portion of graphical code does not execute in response to the first programmatic event being generated;~~

wherein said dynamically registering the first programmatic event causes the first portion of graphical code to execute in response to the first programmatic event being generated.

88. (Previously Presented) The method of claim 87, further comprising:

receiving user input specifying the first programmatic event to be dynamically registered during execution of the graphical program;

wherein the graphical program is configured to dynamically register the first programmatic event during execution of the graphical program in response to the user input specifying the first programmatic event.

89. (Previously Presented) The method of claim 87,

wherein said configuring the graphical program to dynamically register the first programmatic event during execution of the graphical program comprises configuring a node in the graphical program to dynamically register the first programmatic event during execution of the graphical program.